Evaluation and Classification of Computer Architectures

Bewertung und Klassifikation von Rechnerarchitekturen

Architecture

- System Architecture
- Software Architecture
- Hardware Architecture

- Hardware Architecture = Computer Architecture
  - ISA: instruction set architecture
  - HSA: hardware system architecture

ISA: Hardware / Software Interface

Architect

searches the space of possible designs for the best solution (best design).

Goals
Criteria for a good design

Cost / Performance Analysis

Good Ideas

Bad Ideas

Mediocre Ideas

Architectural Goals

- High Performance
- High Dependability
  - reliability, availability, maintainability, security, safety
- Ease of Use
- Extendability
- Scaleability
- Low Costs
Architectural Criteria

**Modularity:** Collection of (nearly) independent modules
- ⇒ extendability, availability

**Orthogonality:** Each module performs a well defined function; no two modules perform the same function
- ⇒ maintainability, costs, usage

**Adequacy:** The costs of a module is adequate with respect to its function and use.
- ⇒ performance, costs

**Symmetry:** Existence of one module implies the existence of certain other modules
- ⇒ ease of use

**Virtuality:** Elimination of physical limits (by software)
- virtual processor, virtual memory, virtual channel, ...
  - ⇒ scaleability, extendability, ease of use

**Transparency:** Hiding of 'unimportant' details from the user
- ⇒ ease of use
Transparency

Transparency of access: identical operations for accessing different media (main memory, mass storage, network,..)
⇒ ease of use

Transparency of location: access of objects without knowledge of their location
⇒ ease of use

Transparency of replication: transparent use of several copies of an object
⇒ performance, dependability

Transparency of concurrency: transparent use of parallelism
⇒ performance, extendability, scaleability

Transparency of performance: system configuration can be changed without loss of functionality, performance, reliability etc.
⇒ extenability, dependability, maintainability

Transparency of scale: system size can be extended such that its performance increases
⇒ scaleability, costs

Transparency of faults: the system hides, masks, tolerates faults and errors
⇒ dependability

Cost/Performance Analysis - Architektur-Analysen

Machbarkeitsstudie / Feasability Analysis

Leistungsbewertung / Performance Analysis

Verläßlichkeitsbewertung / Dependability Analysis

Kostenanalyse / Cost Analysis

E-V-V

Verification / Verifikation: Correctness of the design
Is the system correctly designed and implemented?

Evaluation / Bewertung: What are the parameter values?
What can the system deliver?

Validation / Validierung: Does the system what is expected?
Has the correct system been designed and implemented?

- verification and evaluation of architectures
- validation and evaluation of systems
Bewerten von System- und Rechnerarchitekturen

Ziele / Goals
- identification of weak points in the architecture
- identification of performance bottlenecks
- identification of hidden reserves
- predictions of performance, reliability and costs
- comparison of architectures
- evaluation of new concepts
- tuning for the application
- etc.

Kriterien / Criteria

Bewertungsmaße / Measures

Leistungsgrößen / Performance Measures
- Raten / rates: MIPS, MFLOPS
- CPU-Performance oder mittlerer CPU-Durchsatz (CPU-Throughput)
- Beschleunigung / Speed up
- Summarische Bewertung: Benchmarks

Zuverlässigkeitsgrößen / Dependability Measures
- Fehlerraten / failure rates: FITS
- Funktionswahrscheinlichkeit (Zuverlässigkeit) /reliability
- Verfügbarkeit / Availability
- Sicherheit / Safety
- Robustheit / robustness
Bewertungsverfahren / Means

- Messende Bewertung / Measurements
  - Monitoring - Benchmarks
- Simulative Bewertung / Simulation
- Analytische Bewertung / Analysis
- Wissensbasierte Bewertung / Knowledge based Evaluation (Expertensysteme)

⇒ Simulative und analytische Bewertung sind modell-basiert.

Classification of Computer Architectures

Criteria / Klassifikations-Kriterien

Architecture Classes / Klassen

Criteria

- Organization of the Control and Data Flow
- Address Space Organization
- Use of Physical Memory

Classification (1)

- Dependent Program Control:
  - single, central control unit
- Independent Program Control:
  - parallel machines
Examples: Single Control Architectures

Classification (1)

- Dependent Program Control:
  - Single, central control unit

- Independent Program Control:
  - Parallel machines

Why Parallel Computing?

Cost-performance of serial computers

At lower end performance increases faster than linearly, then the curve saturates and small gains in performance come at tremendous increase in cost.

Why Parallel Computing?

Although speed of computers is steadily increasing, new applications require even higher speed.

Performance of serial computers is beginning to saturate.

A natural way to solve the problem is the usage of an ensemble of processors

⇒ parallel computers
Main Purposes for Parallel Computing

A parallel computer is simply a collection of processors, interconnected in a certain fashion to allow the coordination of their activities and the exchange of data.

Parallel computers need parallel algorithms, i.e. algorithms suitable for implementation on parallel computers.

- Faster solutions
- Solution of large size problems

Flynn’s Classes of Computer Architectures

- SI: Single Instruction Stream
- MI: Multiple Instruction Stream
- SD: Single Data Stream
- MD: Multiple Data Stream

Flynn’s Classes of Computer Architectures

- SISD: Single Instruction Stream—Single Data Stream
  - Monoprocessor
  - Princeton (Von Neumann) - Architecture: ñ common instruction and data stream
  - Harward - Architecture: ñ separate instruction and data streams

- SIMD: Single Instruction Stream—Multiple Data Stream
  - array processors, vector computers
Flynn’s Classes of Computer Architectures

- MIMD: Multiple Instruction Stream-Multiple Data Stream

Massively Parallel Processor: MPP
- High number of processing nodes
- High switching network
- Single computing resource - for a single job

Degree of Parallelism

Moderate Parallel Computers:
- SMP (Symmetric Multiprocessor)

Symmetric Multiprocessor

In einem (symmetrischen) Multiprozessorsystem können alle Prozessoren auf einen gemeinsamen Hauptspeicher zugreifen.

Dadurch ist es ihnen möglich, Daten auszutauschen.

Den Prozessoren steht aber nur ein gemeinsamer Datenweg zum Hauptspeicher zur Verfügung, z.B. ein gemeinsamer Bus.

Non-symmetric Multiprocessor

Haupt- und Ein/Ausgabe-Prozessoren
Parallel Architectures

- Control / Data Flow: centralized or distributed (Flynn)
- Address Space: local or global
- Physical Memory: private or shared
- Type of Interconnection Network